√RADIX

Cerberus - How Radix achieves infinite linear scalability while preserving atomic composability

Summary

2

Why Blockchains Can’t Scale - Episode 3

We begin by outlining some core concepts, such as what nodes and blockchains are.

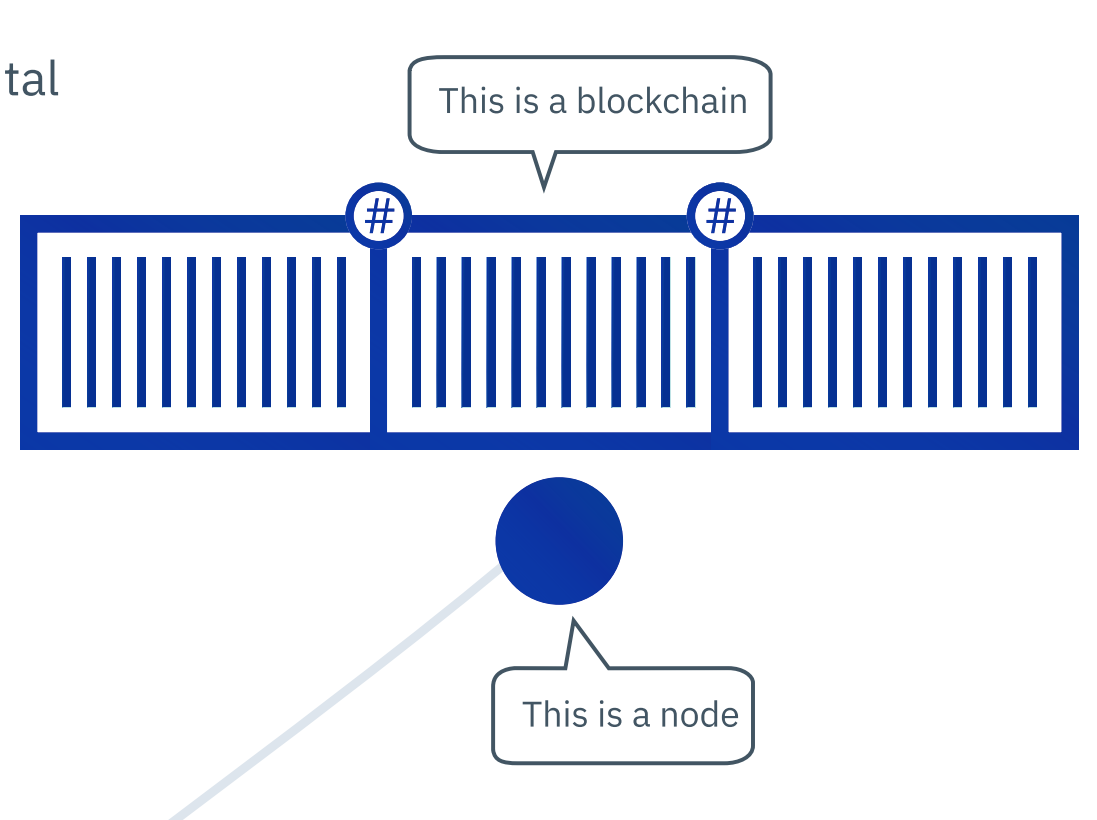
We then explore why blockchains can’t scale, at a fundamental level.

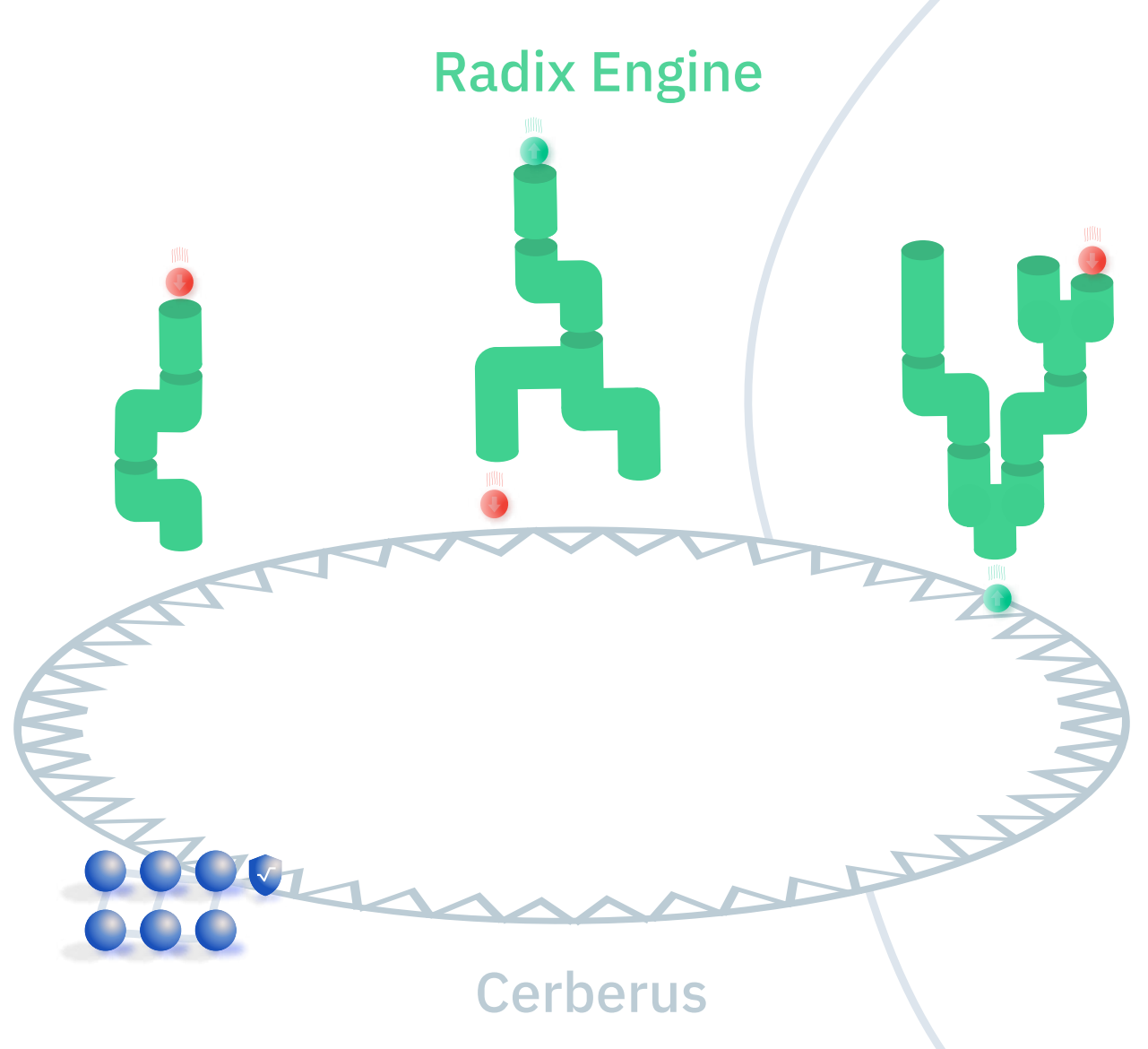
Every available scaling solution tried to date, such as:
- pushing more transactions down the same chain;
- “sharding” or “Layer 2s” solutions;
- centralizing aspects of processing;

either reaches a scalability limit, or sacrifices, to varying extents: decentralization, security, or composability - the ability to settle transactions seamlessly across the ledger.

Blockchains are therefore not the technology that will allow Decentralized Finance (“DeFi”) to scale to fulfill the future needs of billions of people.

A new paradigm is needed.





What is Radix - Episode 4

Radix is a “Layer 1” protocol. This means it’s the foundational software run by nodes that together form the Radix network.

Radix is not a blockchain. Its design is radically different from all other decentralized networks. Radix is unique.

Radix is composed of two layers:
- Radix Engine - for applications;
- Cerberus - for consensus.

Anyone can program rules using the Radix Engine. Those rules govern how assets such as tokens, or decentralized applications, work.

This infographic series is focused on Cerberus, Radix’s consensus layer.

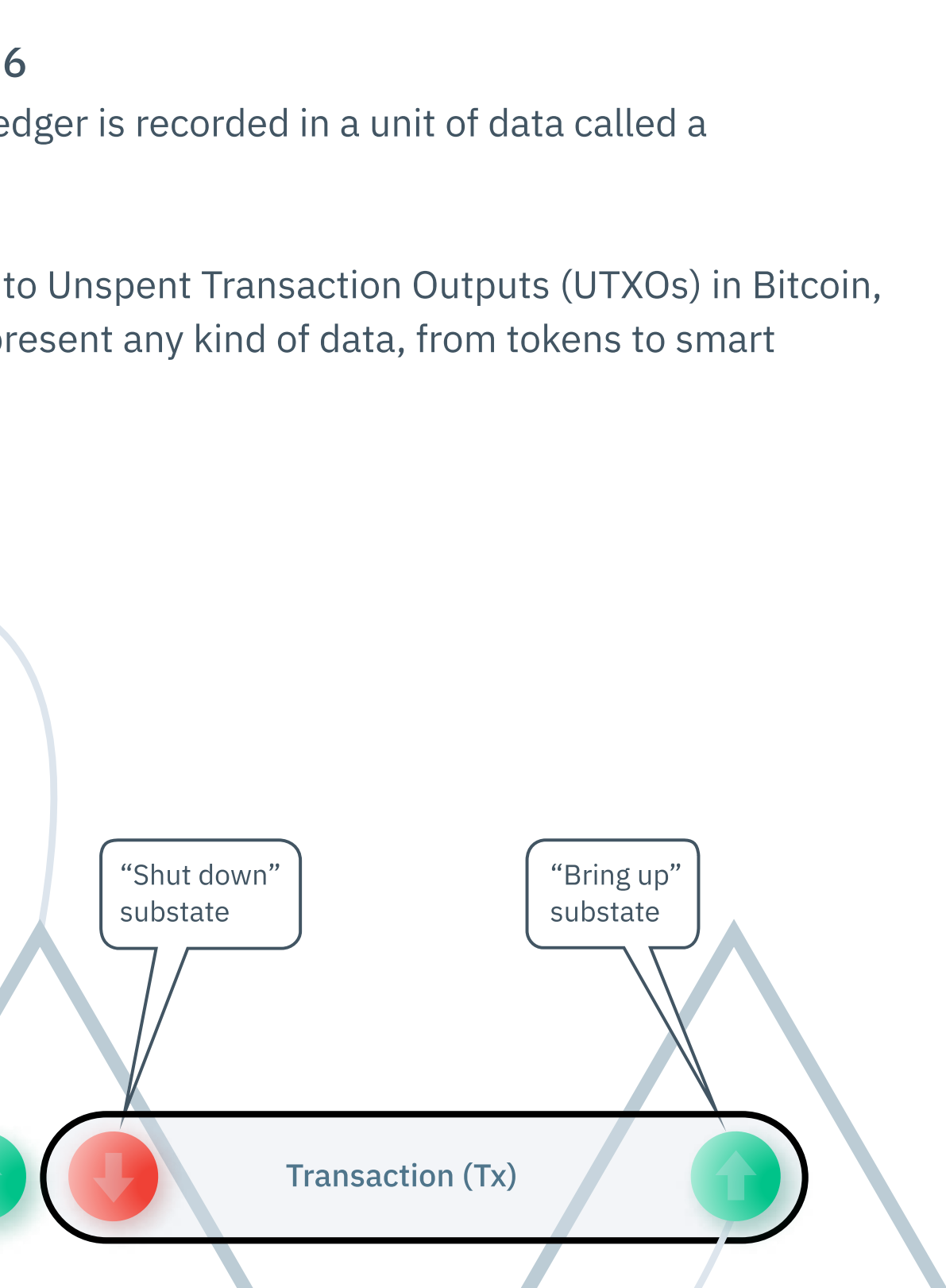
The Shardspace and Validator Sets - Episode 5

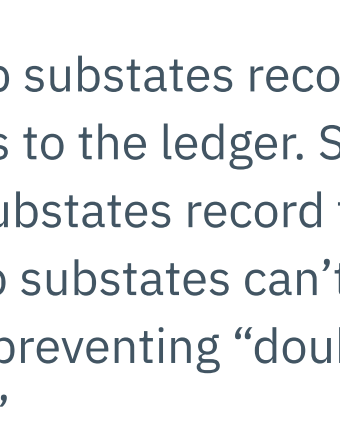
Radix's use of Cerberus as its consensus layer starts by presharding its ledger into 2^{256} shards.

Those shards are fixed, and will never change. All shards put together become the “Shardspace”.

Each individual shard is served by a collection of computers - nodes - called a “Validator Set”.

A validator set is responsible for verifying, voting on, and maintaining a record of transactions for shards they serve.





Substate - Episode 6

Every change to the ledger is recorded in a unit of data called a “substate”.

Substates are similar to Unspent Transaction Outputs (UTXOs) in Bitcoin, but substates can represent any kind of data, from tokens to smart contracts.

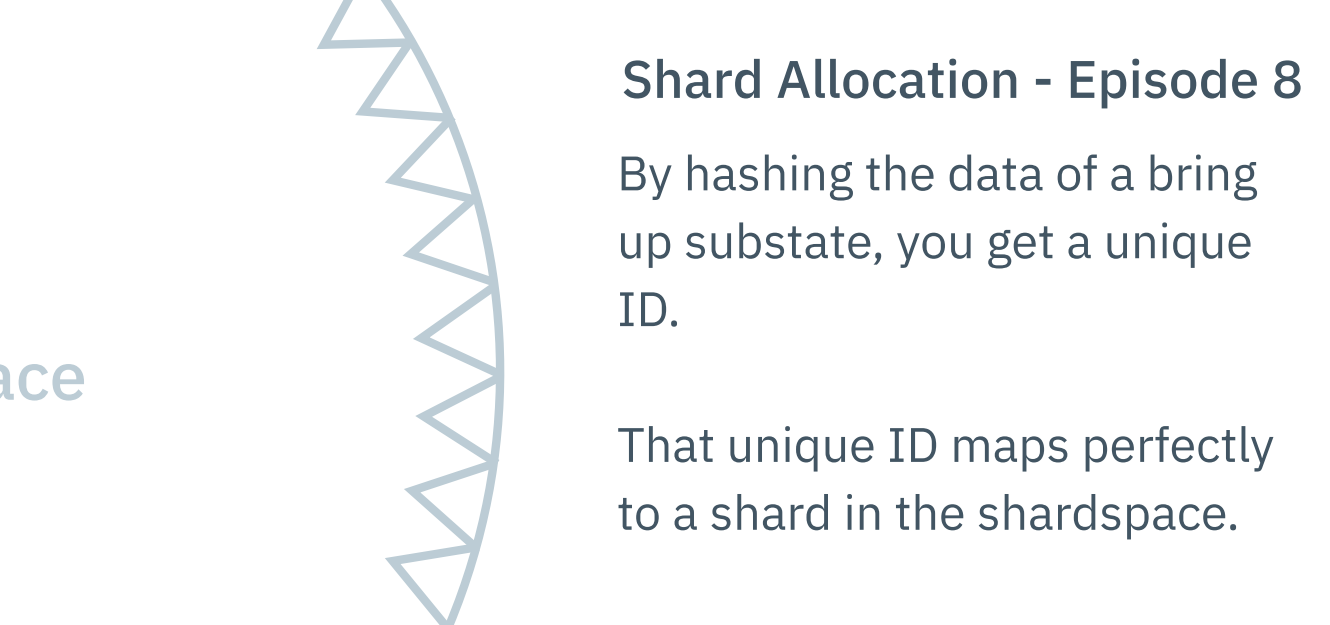
Substate and Transactions - Episode 7

Substate is created in transactions.

Transactions chain together two types of substate: “bring up” and “shut down” substates, across shards.

Bring up substates record changes to the ledger. Shut down substates record that bring up substates can’t be used again - preventing “double spends”.

Each bring up substate has its own shard.



256 bit #

Substate ID: 144,826,489,488,222,149,009,878

Shard Allocation - Episode 8

By hashing the data of a bring up substate, you get a unique ID.

That unique ID maps perfectly to a shard in the shardspace.

Therefore every bring up substate has its own deterministically allocated shard.

Transactions - Episode 9

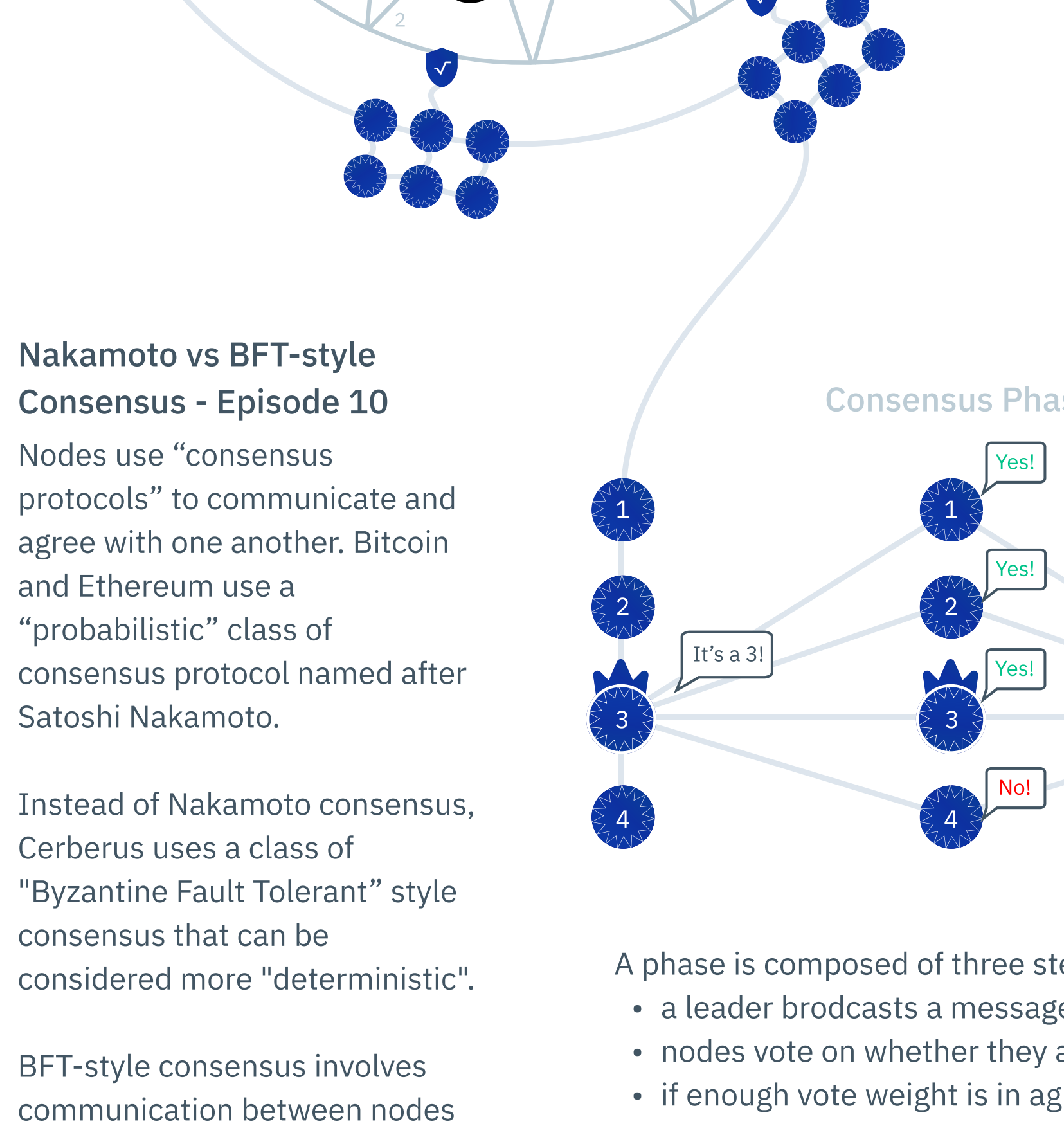
Transactions are instructions sent to nodes to tell them what to write to the ledger.

They contain all the details necessary to write substates to individual shards.

Transaction (Tx)

Shard 1: *Shut down Alice’s Token A substate*
Shard 2: *Bring up Bob’s Token A substate [Index: 0]*
Shard 3: *Shut down Bob’s Token B substate*
Shard 4: *Bring up Carol’s Token B substate [Index: 1]*
Shard 5: *Shut down Carol’s Token C substate*
Shard 6: *Bring up Alice’s Token C substate [Index: 2]*
Signed: *Alice, Bob, Carol*

Tx ID: ...979



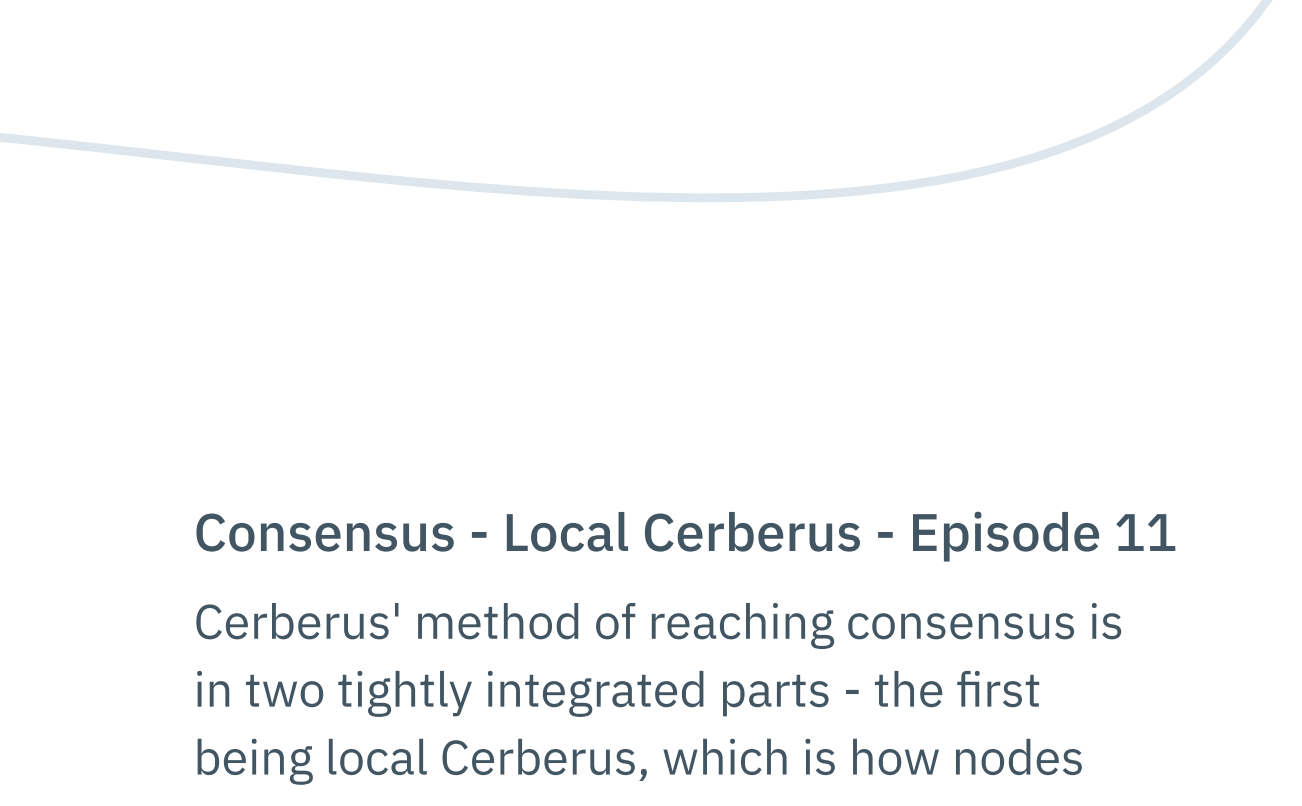
But nodes need a way to agree with one another on whether the transactions that are sent to them are valid, before they can commit the transaction to the ledger.

Nakamoto vs BFT-style Consensus - Episode 10

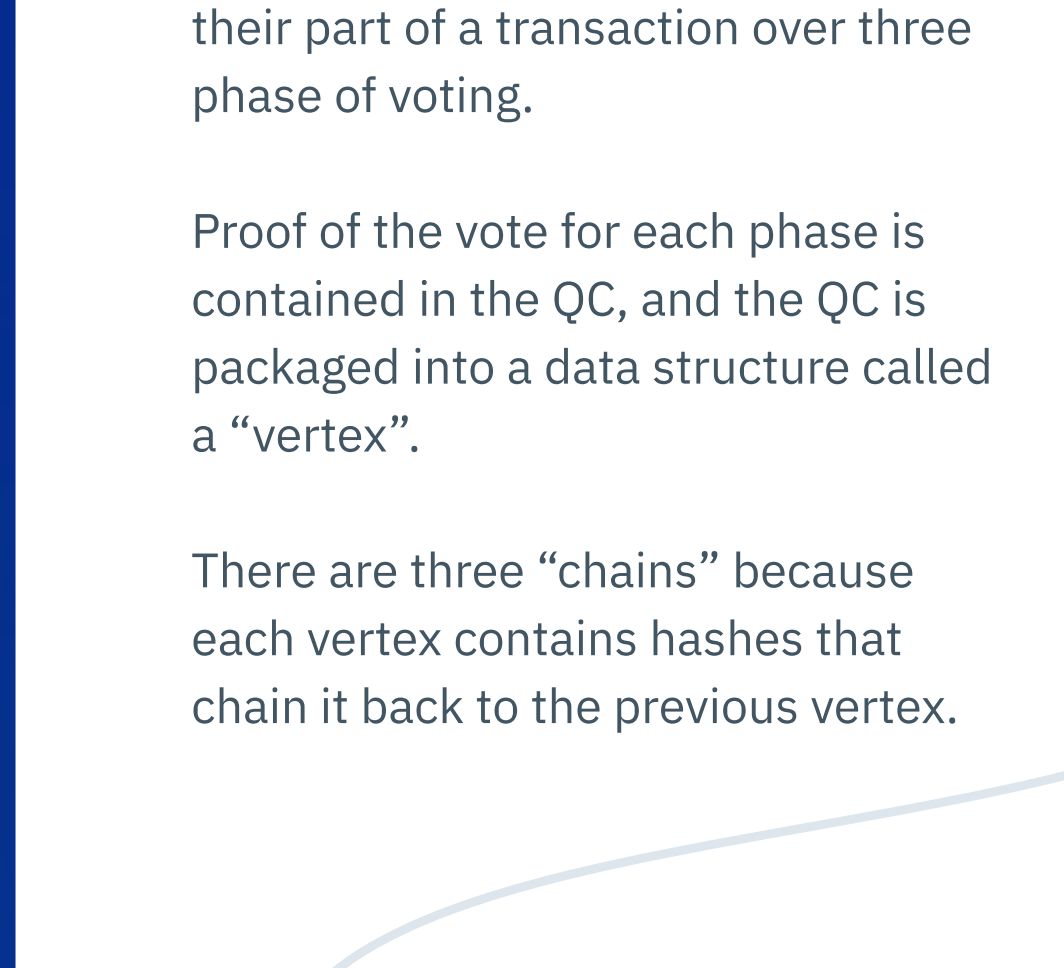
Nodes use “consensus protocols” to communicate and agree with one another. Bitcoin and Ethereum use a “probabilistic” class of consensus protocol named after Satoshi Nakamoto.

Instead of Nakamoto consensus, Cerberus uses a class of "Byzantine Fault Tolerant” style consensus that can be considered more "deterministic".

BFT-style consensus involves communication between nodes in "phases", an example of which is depicted to the right.



A phase is composed of three steps:
- a leader brodcasts a message
- nodes vote on whether they agree with the message
- if enough vote weight is in agreement, the leader creates a Quorum Certificate, QC, which provides cryptographic proof of the outcome of the vote.



Consensus - Local Cerberus - Episode 11

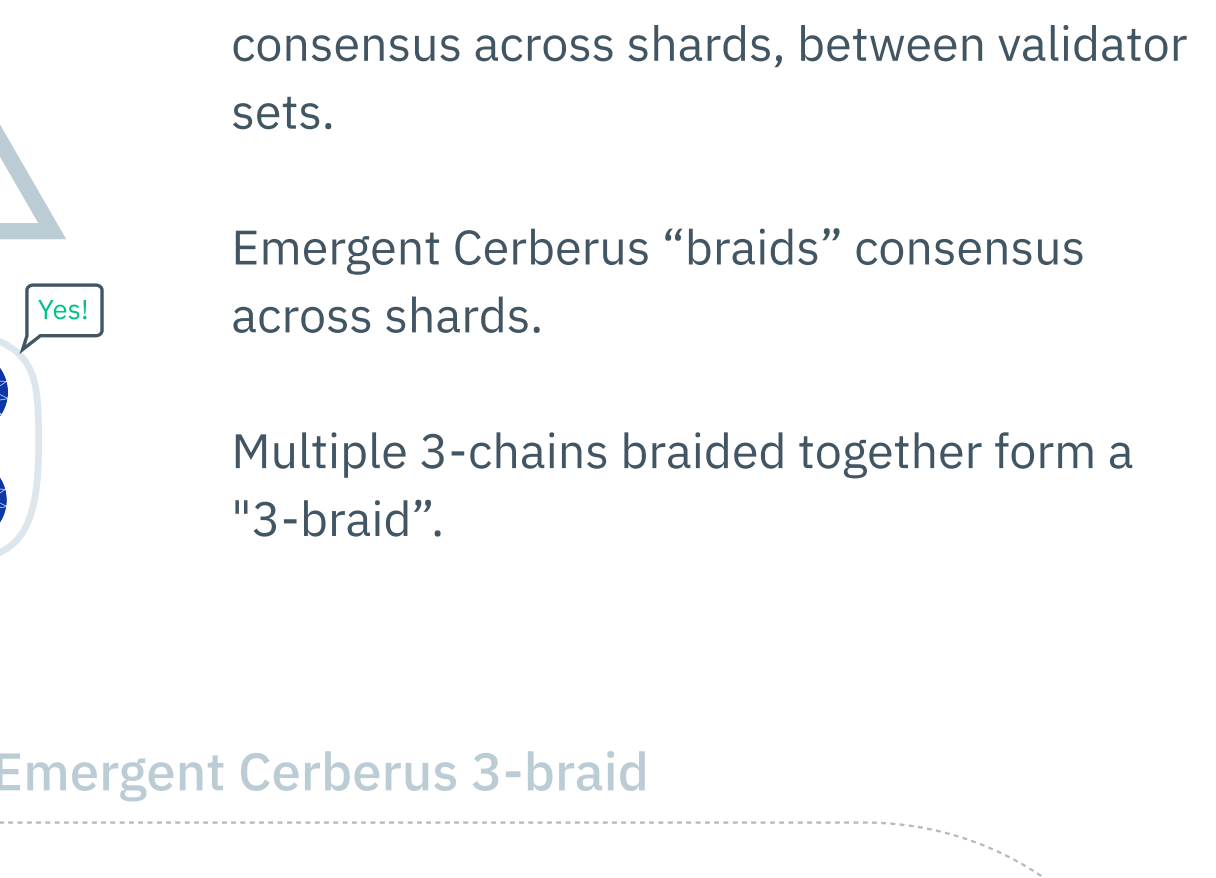
Cerberus' method of reaching consensus is in two tightly integrated parts - the first being local Cerberus, which is how nodes come to consensus with each other, for a single shard, within a validator set.

Consensus in local Cerberus forms a “3-chain”.

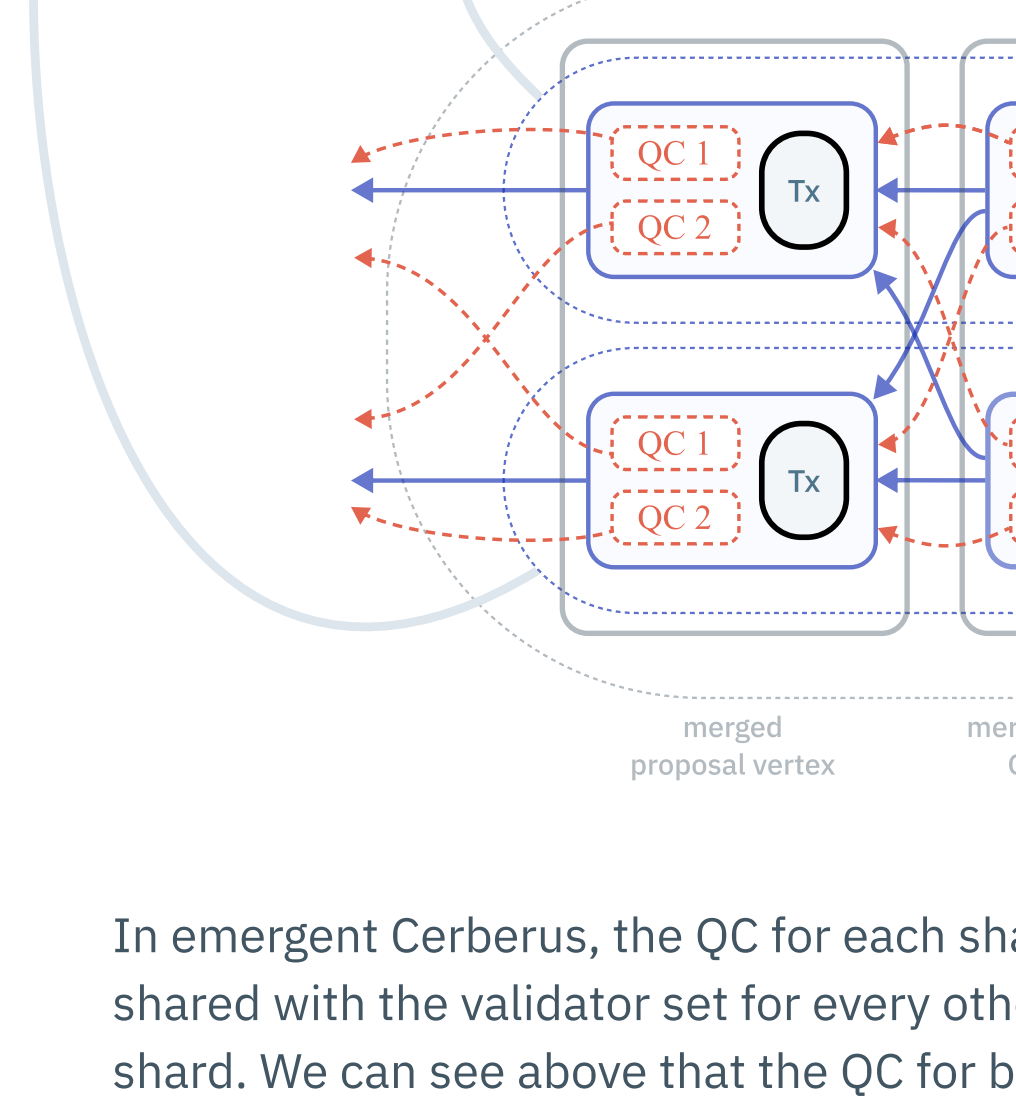
A 3-chain is proof that the nodes of a single shard's validator set approved their part of a transaction over three phase of voting.

Proof of the vote for each phase is contained in the QC, and the QC is packaged into a data structure called a “vertex”.

There are three “chains” because each vertex contains hashes that chain it back to the previous vertex.



These arrows are hashes that tie vertices together. These two hashes tie back to the previous transaction.

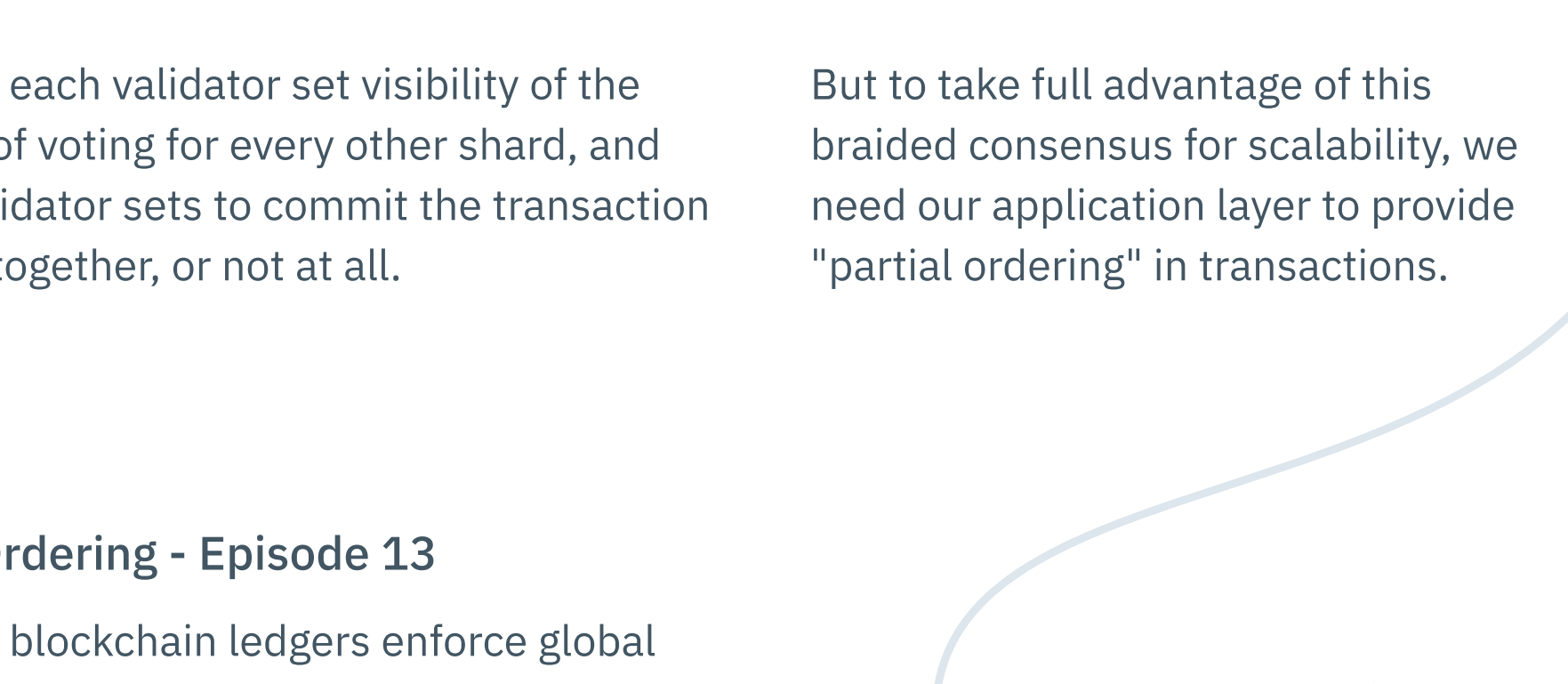


Consensus - Emergent Cerberus - Episode 12

The second part of Cerberus is Emergent Cerberus, which is how nodes come to consensus across shards, between validator sets.

Emergent Cerberus “braids” consensus across shards.

Multiple 3-chains braided together form a "3-braid”.



In emergent Cerberus, the QC for each shard is shared with the validator set for every other shard. We can see above that the QC for both shards 1 and 2 are contained within every vertex for every shard.

This gives each validator set visibility of the outcome of voting for every other shard, and allows validator sets to commit the transaction either all together, or not at all.

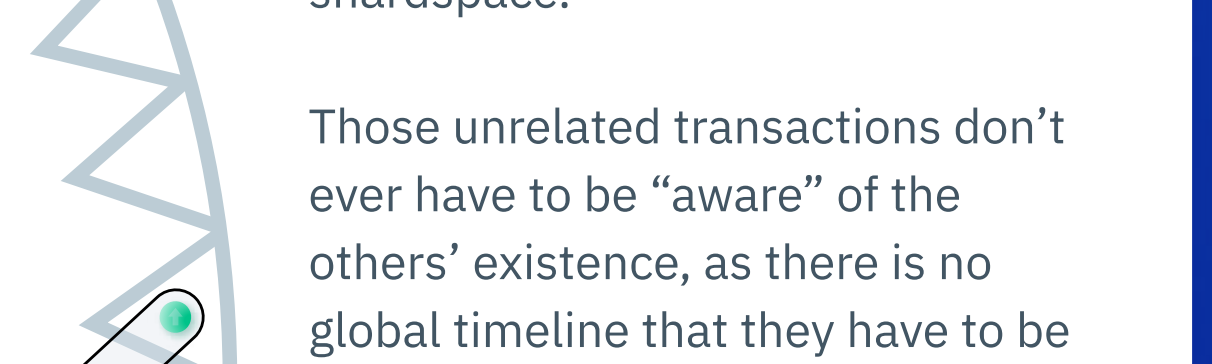
This is the key innovation of Cerberus and uniquely provides Cerberus the ability to settle transactions atomically across shards, also known as “cross-shard atomic composability”.

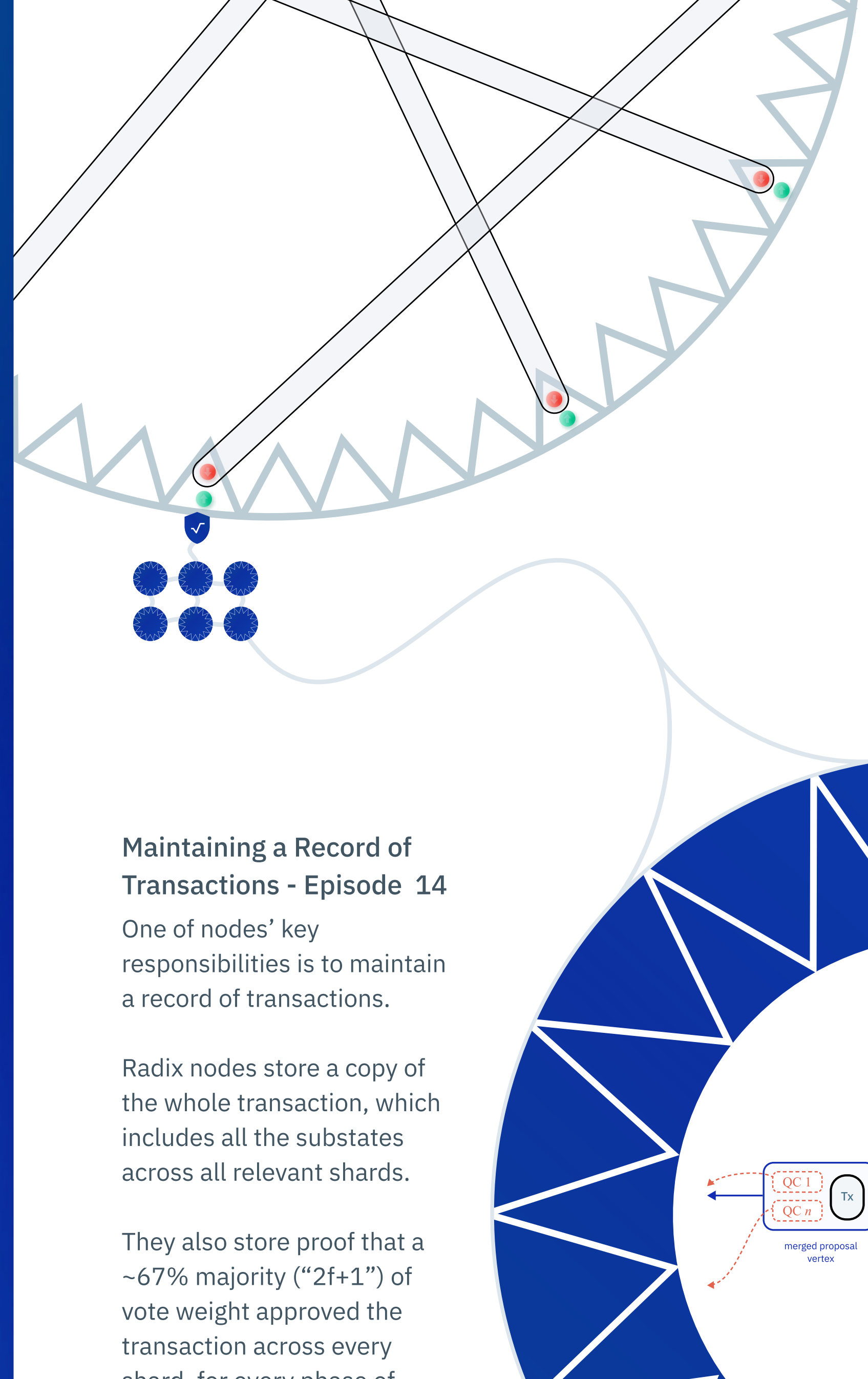
But to take full advantage of this braided consensus for scalability, we need our application layer to provide "partial ordering" in transactions.

Partial Ordering - Episode 13

Nearly all blockchain ledgers enforce global ordering, which is when all transactions are ordered on the same global timeline.

This limits the transaction throughput of those networks as unrelated transactions have to wait for other unrelated transactions before they can be settled.





One of Cerberus' innovations is enabling a ledger that can be "partially ordered”.

Where substates are related, they are composed and ordered within the same atomic transaction.

Where substates are unrelated, they can be processed by transactions completely in parallel, across the vastness of the shardspace.

Those unrelated transactions don’t ever have to be “aware” of the others’ existence, as there is no global timeline that they have to be placed on. As all transactions happen in parallel, no transaction ever has to wait for another transaction as it just gets processed immediately.

This allows Cerberus to scale transaction throughput “linearly”, as every additional node added to the network adds more computation and bandwidth, as the same amount of shardspace is now served by more nodes.

Because the shardspace is practically infinite in size, the linear scalability offered by Cerberus will never hit a ceiling. That’s how Radix achieves “practically infinite linear scalability”.

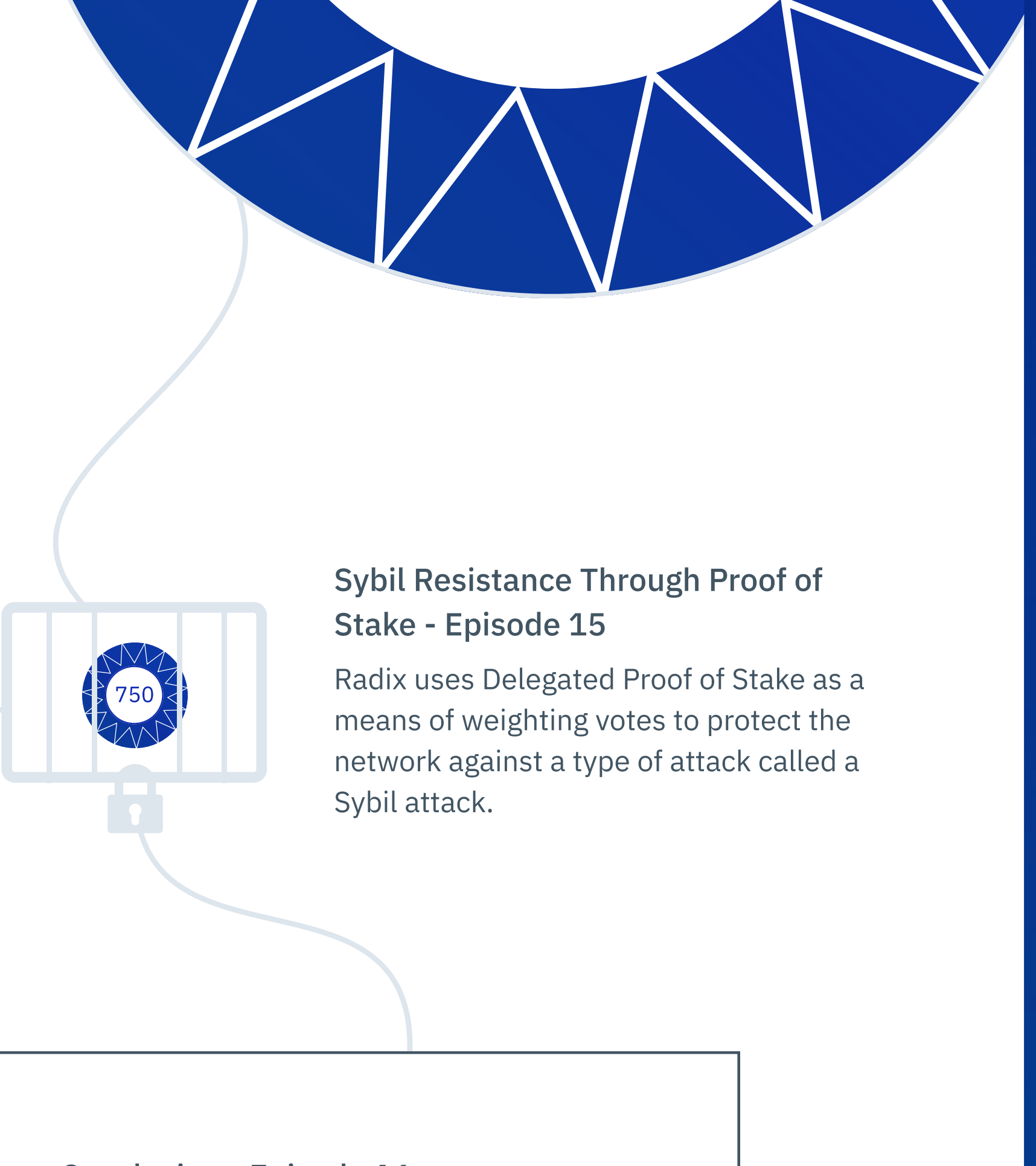
Maintaining a Record of Transactions - Episode 14

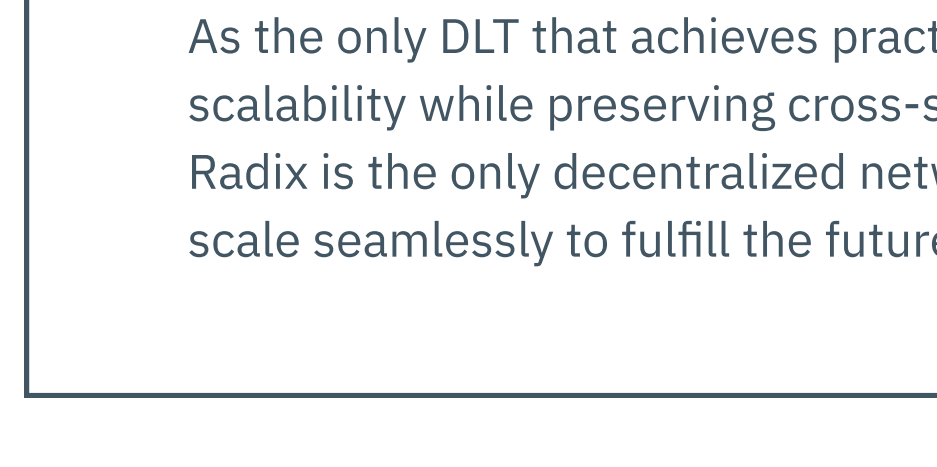
One of nodes’ key responsibilities is to maintain a record of transactions.

Radix nodes store a copy of the whole transaction, which includes all the substates across all relevant shards.

They also store proof that a ~67% majority (“2f+1”) of vote weight approved the transaction across every shard, for every phase of consensus through the QCs in each vertex.

Each vertex is chained to the previous vertex via hashes - cryptographically tying every phase of consensus together.





Sybil Resistance Through Proof of Stake - Episode 15

Radix uses Delegated Proof of Stake as a means of weighting votes to protect the network against a type of attack called a Sybil attack.

Conclusion - Episode 16

We finish by revisiting why blockchains can’t scale, and reflect on how Radix's unique design makes it sui generis: a new paradigm in decentralized DLT.

As the only DLT that achieves practically infinite linear scalability while preserving cross-shard atomic composability, Radix is the only decentralized network that will allow DeFi to scale seamlessly to fulfill the future needs of billions of people.

The version of Cerberus described in this infographic series is scheduled to launch as part of the fully sharded Radix Xi’an release. Please visit www.radixdlt.com for details on the Radix roadmap.

© Copyright Radix Tokens (Jersey) Limited

v1.0 · June 2021